# Package 'RDeco'

December 1, 2016

**Type** Package

**Title** Clusterized and parallelized implmentation of DECO algorithm

**Version** 0.1.0

**Author** J. Carter, S. Davenport, J. Knoblauch, G. Morina

**Maintainer** Sam Davenport <sam.davenport@spc.ox.ac.uk>

**Description** The package provides methods to run the DECO algorithm as described in ``DECOrrelated feature space partitioning
for distributed sparse regression" in Wang, Dunson, and Leng (2016).

**License** Undefined

**LazyData** TRUE

**Imports** Rcpp (>= 0.12.8), RcppArmadillo (>= 0.7.500.0)

**LinkingTo** Rcpp, RcppArmadillo

**Depends** RcppArmadillo, parallel, glmnet

**RoxygenNote** 5.0.1

## R topics documented:

DECO_LASSO_C_PARALLEL . . . . . 1
DECO_LASSO_MIX . . . . . 3
DECO_LASSO_R . . . . . 4
DECO_LASSO_R_CLUSTER . . . . . 5

**Index** **7**

---

DECO_LASSO_C_PARALLEL *DECO Parallelized Algorithm (Pure C++)*

---

### Description

This implements the algorithm DECO which was introduced in "DECOrrelated feature space partitioning for distributed sparse regression" by Wang, Dunson, and Leng (2016). It assumes that we take the lasso to be the penalized regression scehme.

## Usage

```
DECO_LASSO_C_PARALLEL(Y, X, p, n, m, lambda, r_1, r_2 = 0.01, ncores = 1L,
  intercept = TRUE, refinement = TRUE, glmnet = TRUE,
  parallel_glmnet = FALSE, precision = 1e-07, max_iter = 100000L)
```

## Arguments

| | |
|---|---|
| Y | gives the nx1 vector of observations we wish to approximate with a linear model of type Y = Xb + e. |
| X | gives the nxp matrix of regressors, each column corresponding to a different regressor. |
| p | is the column dimension of X [equivalently, p is the number of regressor variables]. |
| n | is the row dimension of X (and Y) [equivalently, n is the number of observations/individuals]. |
| m | is the number of groups/blocks you wish to split X into, denoted X(i) for 1 <= i <= m. |
| lambda | gives the (fixed) penalty magnitude in the LASSO fit of the algorithm. |
| r_1 | is a tweaking parameter for making the inverse more robust (as we take inverse of XX + r_1*I). |
| r_2 | is a tweaking parameter for making the inverse more robust (as we take inverse of X_MX_M + r_2*I). |
| ncores | determines the number of threads used on each machine to parallelize computation. |
| intercept | determines whether to include an intercept in the model or not. |
| refinement | determines whether to include the refinement step (Stage 3 of the algorithm). |
| glmnet | determines whether glmnet function form glmnet R package should be used to compute the Lasso coefficients. See details for further information. If set to FALSE, C++ implementation of coordinate descent algorithm is used. |
| parallel_glmnet | determines whether a parallel version of the Lasso coefficients should be used. This parameter is ignored when glmnet is set to FALSE (see details). |
| precision | determines the precision used in the coordinate descent algorithm. It is ignored when glmnet is set to TRUE. |
| max_iter | determines the maximum number of iterations used in the coordinate descent algorithm. It is ignored when glmnet is set to TRUE. |

## Details

This function is a C++ implementation of DECO_LASSO_R and DECO_LASSO_MIX functions. Due to the fact that it is entirely written in C++ it runs faster than the corresponding R implementations for sufficiently large matrices.

Two functions can be used to compute Lasso coefficients: glmnet R function (glmnet = TRUE). and coordinate descent algorithm (glmnet = FALSE). glmnet R function is generally faster, but more memory is required to pass the input argumentd from C++ to R and back. When parallel_glmnet = TRUE an R parallelized version of glmnet is used. Note however that for small datasets this could lead to slower run times, due to the communication between C++ and R.

Descent coordinate algorithm is always run in a parallel way (using ncores threads).

## Value

An estimate of the coefficients b.

## Author(s)

Samuel Davenport, Jack Carter, Giulio Morina, Jeremias Knoblauch

---

`DECO_LASSO_MIX` *DECO Parallelized Algorithm (Mixture of R and C++)*

---

## Description

This implements the algorithm DECO using a mixture of R and C++ code.

## Usage

```
DECO_LASSO_MIX(Y, X, p = NULL, n = NULL, m = 1, lambda, r_1, r_2 = r_1,
  ncores = 1, intercept = TRUE, refinement = TRUE)
```

## Arguments

| | |
|---|---|
| `Y` | gives the nx1 vector of observations we wish to approximate with a linear model of type Y = Xb + e. |
| `X` | gives the nxp matrix of regressors, each column corresponding to a different regressor. |
| `p` | is the column dimension of X [equivalently, p is the number of regressor variables]. If not given, it is computed as the number of columns of X. |
| `n` | is the row dimension of X (and Y) [equivalently, n is the number of observations/individuals]. If not given, it is computed as the number of rows of X. |
| `m` | is the number of groups/blocks you wish to split X into, denoted X(i) for 1 <= i <= m. |
| `lambda` | gives the (fixed) penalty magnitude in the LASSO fit of the algorithm. |
| `r_1` | is a tweaking parameter for making the inverse more robust (as we take inverse of XX + r_1*I). |
| `r_2` | is a tweaking parameter for making the inverse more robust (as we take inverse of X_MX_M + r_2*I). |
| `ncores` | determines the number of threads used on each machine to parallelize computation. |
| `intercept` | determines whether to include an intercept in the model or not. |
| `refinement` | determines whether to include the refinement step (Stage 3 of the algorithm). |

## Details

This implements the algorithm DECO which was introducted in "DECOrrelated feature space partitioning for distributed sparse regression" by Wang, Dunson, and Leng (2016). It assumes that we take the lasso to be the penalized regression scehme.

## Note

-This implementation uses both R functions and C++ functions. In particular, `standardizeMatrix`, `invSymmMatrix`, `squareRootSymmetric` functions are used when needed in place of native R functions. Higher speed can be achieved by using other functions provided in the package.

-This implementation is suboptimal in that X is already stored in the memory when we start the procedure. Ideally, one would give in only the LOCATION X is stored at and read it in chunkwise (thus allowing for larger matrices X, as was intended by the authors).

## Author(s)

Samuel Davenport, Jack Carter, Giulio Morina, Jeremias Knoblauch

---

`DECO_LASSO_R` *DECO Parallelized Algorithm (Pure R)*

---

## Description

This implements the algorithm DECO which was introducted in "DECOrrelated feature space partitioning for distributed sparse regression" by Wang, Dunson, and Leng (2016). It assumes that we take the lasso to be the penalized regression scehme.

## Usage

```
DECO_LASSO_R(Y, X, p = NULL, n = NULL, m = 1, lambda, r_1, r_2 = r_1,
  ncores = 1, intercept = TRUE, refinement = TRUE)
```

## Arguments

| | |
|---|---|
| `Y` | gives the nx1 vector of observations we wish to approximate with a linear model of type Y = Xb + e. |
| `X` | gives the nxp matrix of regressors, each column corresponding to a different regressor. |
| `p` | is the column dimension of X [equivalently, p is the number of regressor variables]. If not given, it is computed as the number of columns of X. |
| `n` | is the row dimension of X (and Y) [equivalently, n is the number of observations/individuals]. If not given, it is computed as the number of rows of X. |
| `m` | is the number of groups/blocks you wish to split X into, denoted X(i) for 1 <= i <= m. |
| `lambda` | gives the (fixed) penalty magnitude in the LASSO fit of the algorithm. |
| `r_1` | is a tweaking parameter for making the inverse more robust (as we take inverse of XX + r_1*I). |
| `r_2` | is a tweaking parameter for making the inverse more robust (as we take inverse of X_MX_M + r_2*I). |
| `ncores` | determines the number of threads used on each machine to parallelize computation. |
| `intercept` | determines whether to include an intercept in the model or not. |
| `refinement` | determines whether to include the refinement step (Stage 3 of the algorithm). |

## Details

The algorithm is based on the description in "DECOrrelated feature space partitioning for distributed sparse regression" in Wang, Dunson, and Leng (2016) if lambda is fixed and LASSO is used as the penalized regression scheme. The rotated versions of Y and X the authors denote with Tilde are denoted as X* and Y* in the comments below

## Note

-This implementation uses only R functions. Higher speed can be achieved by using other functions provided in the package.

-This implementation is suboptimal in that X is already stored in the memory when we start the procedure. Ideally, one would give in only the LOCATION X is stored at and read it in chunkwise (thus allowing for larger matrices X, as was intended by the authors).

-The notation #~PARALLEl~# will be introduced in the code whereever one may achieve signficiant gains from parallelizing

-I could evaluate old expressions in the R version within the mcapply loops! ->saves memory as we write over old data

-We cannot disturb variable order within the algorithm for output comparison reasons, thus reorder X columns before running DECO_LASSO (if important)

## Author(s)

Samuel Davenport, Jack Carter, Giulio Morina, Jeremias Knoblauch

---

# DECO_LASSO_R_CLUSTER *DECO Clusterized Algorithm (Pure R)*

---

## Description

DECO Clusterized Algorithm (Pure R)

## Usage

```
DECO_LASSO_R_CLUSTER(Y, X, p, n, lambda, r_1, clust, r_2 = r_1, ncores = 1,
  intercept = TRUE, refinement = TRUE)
```

## Arguments

| | |
|---|---|
| Y | gives the nx1 vector of observations we wish to approximate with a linear model of type Y = Xb + e |
| X | gives the nxp matrix of regressors, each column corresponding to a different regressor |
| p | is the column dimension of X [equivalently, p is the number of regressor variables]. |
| n | is the row dimension of X (and Y) [equivalently, n is the number of observations/individuals]. |
| lambda | gives the (fixed) penalty magnitude in the LASSO fit of the algorithm |

| | |
|---|---|
| r_1 | is a tweaking parameter for making the inverse more robust (as we take inverse of XX + r_1*I) |
| clust | an object obtained by makePSOCKcluster |
| r_2 | is a tweaking parameter for making the inverse more robust (as we take inverse of X_MX_M + r_2*I) |
| ncores | determines the number of threads used on each machine to parallelize computation |
| intercept | determines whether to include an intercept in the model or not |
| refinement | determines whether to include the refinement step (Stage 3 of the algorithm) |

## Details

The algorithm is based on the description in "DECOrrelated feature space partitioning for distributed sparse regression" in Wang, Dunson, and Leng (2016) if lambda is fixed and LASSO is used as the penalized regression scheme. The rotated versions of Y and X the authors denote with Tilde are denoted as X* and Y* in the comments below

## Note

-This implementation uses only R functions.

- This implementation is meant to distribute the load of work to several machines. Note that the current implementation does not deal with the problem of storing big matrices; this function is just the starting step and it should be further developed (i.e. reading the matrix chunckwise from a file, C++ implementation,parallelizing on each machine,...).

## Author(s)

Samuel Davenport, Jack Carter, Giulio Morina, Jeremias Knoblauch

# Index

DECO_LASSO_C_PARALLEL, 1
DECO_LASSO_MIX, 3
DECO_LASSO_R, 4
DECO_LASSO_R_CLUSTER, 5